

# Implementation and Comparison of the Way That Office Document Is Converted to PDF Documents in the Java Runtime Environment

Jiang Lizheng, Xin Zheng, and Zhao Jiantao

**Abstract**—Java is one of the mainstream programming languages in current. Well, the traditional way that Java calling the COM component of office document is very complicated and may lead to some bugs because of the problem cause by data structure. This paper has implemented and analysed three mainstream ways to convert the document and indicate their efficiencies through the experiments. Finally, we found the best way in order to learn and use.

**Index Terms**—Java, office document, JNI, PDF document, automation.

## I. INTRODUCTION

Portable Document Format, abbreviated PDF, translated into a portable format. Because of its excellent portability, cross-platform, use of encryption features, preview and a variety of out targets, it has become the facts of the current distribution and exchange of electronic document. Therefore, the report is stored in PDF format will help its broader application and dissemination.

Java is one of the mainstream programming languages and has the characteristics of cross platform. Well, these characteristics make it hard for java to call the local function. Although we can call the local function through JNI technology, its complexity weakened the convenience of Java language. This paper has implemented and discussed three ways to convert the office document to PDF document in the Java environment. A further comparison of the advantages and disadvantages of these ways and show the conversion efficiency through Experimental data. Which allow readers to choose according to their needs in their own way. It is also an example to call a COM component in Java programs [1].

## II. JNI

From the beginning of the Java1.1, the standard of Java Native Interface (JNI) has become a part of the Java platform. It allows Java code to interact with other language's code. JNI

started to design for local compiled language, especially for C and C++. But it does not preclude other languages, as long as the calling convention supported on it. Using java code to interact with the local compiled code usually lost platform portability. However, in some cases this is acceptable. For example, the use of some of the old library, interact with the hardware or operating system, or in order to improve the performance of our program. JNI standards ensure local code can work in any Java virtual machine environment. The schematic connecting between Java and C programs through JNI is shown in Fig. 1.

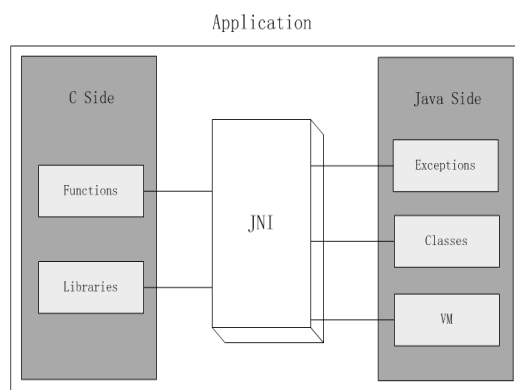


Fig. 1. Schematic diagram of JNI principle.

The specific implementation process is as follows:

- 1) Statement native method in Java, and then compile it.
- 2) generate a .h file through javah command;
- 3) Write a .cpp file to export to achieve native method, which needs to include the .h file generated by the second step (which should also contains a file jni.h in JDK files);
- 4) compiled in the third step of the .Cpp file into a dynamic link library file;
- 5) Load in Java using System.loadLibrary () method generated by the fourth step produces a dynamic link library files which make the native () method can be used in java;

As can be seen from the above process, the way use JNI is too complex and some C programming skills should be required.

## III. COM COMPONENTS

COM Components is a Technology which is developed by Microsoft Software in order to make programming more in line with human behavior. In the COM architecture, people can develop a variety of components which have specific function. Then they will be in accordance with the need to

Manuscript received May 4, 2015; revised November 16, 2015.

Jiang Lizheng is with the Network Information Engineering, North China Electric Power University(NCEPU), Beijing, Changping 102200 China (e-mail: jianglizheng@epu.com).

Xin Zheng is with North China Electric Power University (NCEPU), Beijing, Changping 102200 China (e-mail: 379137847@qq.com).

Zhao Jiantao is with the Computer Information System Security Engineering, North China Electric Power University (NCEPU), Beijing, Changping 102200 China (e-mail: zhaojiantao@epu.com).

combine for an application of a complex system. The resulting benefits are as follows: 1) System components can be replaced with a new one in order to keep the system upgrade and customization. 2) The same component can be reused in multiple application systems. 3) Application systems can easily be extended to the network environment. 4) The characteristic of COM component has no connection with programming languages and platforms which make all programmers can give full play to their talents and expertise to write module [2].

COM is a method for developing software components. Components are same small binary executable program, they can be used in application, operating system, and other components [3]. The way develop custom COM components like development of dynamic object-oriented API. Multiple COM objects can be connected together to create an application or component system [4]. And components also can be removed or replaced at running time. Many of Microsoft technologies such as ActiveX, DirectX COM and OLE, etc. are based on COM components. And many developers from Microsoft use of COM components to customize their applications and operating system.

#### IV. THE SOLUTION

##### A. The Way to Convert the Document with OpenOffice + pdf2swf

###### 1) Solution 1

This kind of solution based on using JodConverter to link native program to convert documents. JodConverter is a Java Open Document file converter and can be used in many file formats. It uses Open office to do the conversion work.

###### 2) JodConverter Deployment Mode and the Core Code

To download the Jod Converter, you can get it through the web site <http://sourceforge.net/projects/jodconverter/files/>. Keep the file in the Lib directory under the jar package which you had download into the Java Web project in WEB-INF/lib. At the same time you should ensure that have OpenOffice software installed on your computer. The Key code used to convert program file as follows:

```

OpenOfficeConnection connection = new
SocketOpenOfficeConnection(8100);
//Create an object of Openoffice by the class of
SocketOpenOfficeConnection, and defines the connection
time is less than 8100 milliseconds.
connection.connect();
//Connect and start.
DocumentConverter converter = new
OpenOfficeDocumentConverter(connection);
//Create a connection to object transform.
converter.convert(sourceFile, pdfFile);
//Through the convert () method to realize file
conversion.SourceFile instead of the full path of the Office
documents and pdfFile instead of the full path of the
converted PDF file.
pdfFile.createNewFile();
//Create a PDF file

```

```

connection.disconnect();

```

```

//Connection release

```

After five tests of conversion to this project we found that it can reach the average time of 4012 milliseconds in a 40 page PPT file and the average time of 2048 milliseconds in a 38 page WORD file. The speed of conversion is acceptable, but the quality is not good. Although all elements of the word file are converted, some of the elements in PPT file can't change or have some conversion errors. In conclusion, this solution can only be used for small parts of the document and can't really solve the problem.

##### B. The Way to Convert the Document with Adobe Acrobat 8 Professional + Microsoft Office

###### 1) Solution 2

This kind of solution based on an open source project named JCom which can help to convert the files. JCom is the abbreviation of Java-COM Bridge. As the name implies, JCom is equivalent to a bridge which will be able to make Java code directly calling other program and make it easy to access to COM components on the Windows platform, so as to complete the transformation work. The principle of this solution is that we should use a registered and installed Acrobat 8 Professional which there is a function added in the Office program that you can make files save as PDF document. By JCom our Java program can call this function and automatically converted document to a PDF document.

###### 2) JCom deployment mode and the core code

To download the JCom package, you can through the web site <http://sourceforge.net/projects/jcom/>. Extract the JCom package after it has been download. Put the jar package in the JCom into WEB-INF/lib of your Java Web project and at the same time to put jcom.dll into your JDK bin directory or in the folder named windows (It is Recommended to put it into the bin directory of JDK).It is also very important to ensure your computer contains a installed and registered software named Acrobat 8 Professional. The key code of the program and the explanation is as follows:

```

ReleaseManager rm = new ReleaseManager();
//ReleaseManager is equivalent to a container. You can use
it to interact with all JCOM on your machine. According to
the parameters you preach it can use all components JCOM
which can be operated on your machine.
IDispatch xlsApp=null;
//IDispatch is an interface which exposed by the OLE
automation protocol and it can be used to connect the Java
program and Application.
xlsApp = new IDispatch(rm, "Excel.Application");
//To obtain Excel's API (if processing is Word we will
need "Word.Application" to instead of it)
xlsApp.put("Visible", new java.lang.Boolean(false));
//Set whether it is can be visible in the conversion process
of Excel.
app.method("CreatePDF", new Object[] { sourceFile,
pdfFile});
//Through the parameter "CreatePDF" in the method () to
achieve the conversion.SourceFile instead of the full path of
the Office file and pdfFile instead of the full path of the
converted PDF file

```

```
rm.release();
//Connection release
```

After five tests of conversion to this project we found that it can reach the average time of 19244 milliseconds in a 40 page PPT file and the average time of 12048 milliseconds in a 38 page WORD file. Obviously, the computation time is greatly increased. On the other hand, judging from the quality of conversion, some color difference, while the basic elements are transformed well. So it can be meet the needs of the general transformation and dissemination of print. But the working time is 5 times slower than the solution 1, is very difficult for the actual project. Although this is not very successful, it has brought an idea to solve the problem. We can use bridging technology and to use other software to realize file conversion. According to this idea we eventually found a better solution.

### C. The Way to Convert the Document with Microsoft Office 2010 or WPS 2012

#### 1) Solution 3

This solution's idea is very similar to the above solution, but it is come out use JACOB technology. JACOB is also a bridging technology of Java-COM.

There are two main mechanisms which are early binding and late binding that can call COM component.

Early binding, which explicit statement object, data type and so on. The compiler will get enough information in order to links and optimized at compile time, which will bring better performance. To understand this mechanism you can refer to the IBM RJC project which provides a high performance solution. It needs to understand more details of the COM component. Although the framework has been completed most of the Bridge code generation, the overall coding workload is too large and coding relatively has high difficulty. The biggest weakness is RJC only supports some Table interface which provides by early bound COM API.

However, late binding is realized through IDispatch interface. It is as same as the reflection mechanism which Java has used, you can call the method by name or ID. The purpose of this design is to support the operation of the COM through scripting language. Because the script is interpreted and do not have pointers and VTable mechanism which can be found in C++. The disadvantage of this approach is the poor performance in some certain situation. Because it call the object through name or ID at runtime. So we can not know which object and method should be called until runtime. This will inevitably bring some overhead and delay. But the advantages of this approach are also very clear like simple, flexible and need not know the details of the dynamic link library which means we can quickly complete the coding work.

The JACOB project provides an implementation of automation server within the JVM. The core technology is based on Variant and Dispatch interface which implement by JNI. Its design reference to the general automation server which contain by Microsoft C++. While Microsoft's implementation only supports its own virtual machine. Through the JACOB, it is easy to call the late bound in Java programs [4]. The schematic diagram is shown in Fig. 2.

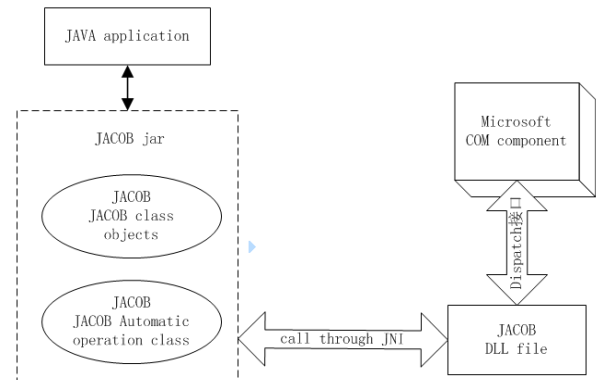


Fig. 2. The working principle that Java program calling COM by JACOB.

#### 2) JACOB Deployment Mode and the Core Code

To download the JACOB package, you can through the web site [http://sourceforge.jp/projects/sfnet\\_jacob-project/](http://sourceforge.jp/projects/sfnet_jacob-project/) to download and jacob-1.18-M2 is the latest version. Put the jar package in the JACOB into WEB-INF/lib of your Java Web project and at the same time to put dll file into your JDK bin directory or in the folder named windows (It is Recommended to put it into the bin directory of JDK). Attention is required, with two DLL files, jacob-1.18-M2-x86.dll and jacob-1.18-M2-x64.dll respectively correspond to the 32 and 64 bit systems. At the same time to ensure that the machine has an Microsoft Office 2010 or Microsoft office 2007 which of has integrated the transformation function of PDF (Because of in 07 conversion this function was just a plug-in, with the popular of PDF, version 10 has been the default integration). The key code of the program and the explanation are as follows:

```
ComThread.InitSTA();
//Initialize the COM thread
ActiveXComponent app = new
ActiveXComponent("Powerpoint.Application" );
//You can also use the CLSID which registered on the local
computer to replace the String "Powerpoint.Application".
Dispatch presentations =
app.getProperty("Presentations").toDispatch();
//To initialize the dispatch interface provided by PPT
program in order to use it later.
Dispatch presentation = Dispatch.call(presentations,
"Open", //open the document
source, // full path of the Office file
true, // ReadOnly
true, // set whether the file should have title
false // set whether the PPT program can be seen
).toDispatch();
Dispatch.call(presentation,
"SaveAs", //The method provided by JACOB used to
save as other format.
pdfFile, // The full path of the PDF file after
conversion
ppSaveAsPDF); //Constants defined in the program.
Dispatch.call(presentation, "Close");
//Close the Dispatch interface
ComThread.Release();
//Release the COM thread
```

After five conversions test to this project we found that it can reach the average time of 4353 milliseconds in a 40 page

PPT file and the average time of 2511 milliseconds in a 38 page WORD file. It has cost similar time comparison with the first solution. In aspect of the quality of conversion, because it is converted by Microsoft office, its transformation has few distortion [5]. Whether the conversion time or quality can meet the actual needs perfectly. You can get the number of PDF file through iText package. We can get some information by the get method after package the program into interface. Through the interface, information has display as in Fig. 3 (To some complex PPT the conversion time will increases)

```

Towcat
Convert document to PDF document...c:\pdfFile\JAUAt.pdf
Document conversion is complete..time cost:11547.ms
Close the file
The total number of pages of the PDF document:80
Return results with Json: {"fid":"123","pagenum":"80","pdfurl":"c:\pdfFile\\","pdfname":"JAUAt.pdf","status":"1"}
Start PPT...
Open file...c:\pdfFile\SWOT.ppt
Convert document to PDF document...c:\pdfFile\SWOTt.pdf
Document conversion is complete..time cost:8187.ms
Close the file
The total number of pages of the PDF document:20
Return results with Json: {"fid":"123","pagenum":"20","pdfurl":"c:\pdfFile\\","pdfname":"SWOTt.pdf","status":"1"}
Start PPT...
Open file...c:\pdfFile\Testt.ppt
Convert document to PDF document...c:\pdfFile\Testt.pdf
Document conversion is complete..time cost:547.ms
Close the file
The total number of pages of the PDF document:1
Return results with Json: {"fid":"123","pagenum":"1","pdfurl":"c:\pdfFile\\","pdfname":"Testt.pdf","status":"1"}
    
```

Fig. 3. The background data of server.

After converting, the file is showing in Fig. 4.

JAVA.ppt	1,035 KB	Microsoft Power...	2015-2-12 10:31	A
JAVAt.pdf	970 KB	Adobe Acrobat D...	2015-11-6 21:21	A
SWOT.ppt	719 KB	Microsoft Power...	2015-2-12 10:31	A
SWOTt.pdf	2,558 KB	Adobe Acrobat D...	2015-11-6 21:22	A
Testt.ppt	31 KB	Microsoft Power...	2015-3-10 15:13	A
Testt.pdf	45 KB	Adobe Acrobat D...	2015-11-6 21:22	A

Fig. 4. The converted files on the server.

## V. CONCLUSION

In this paper, we discuss the problem that convert office document into PDF document. Through comparison of the time cost and the quality of the conversion, we found the best solution. At the same time, it can also be used as a practical example to show how to bypass the JNI to connect to the Java programs and COM components. Finally, we realize the best solution as an interface and deploy it on the server to verify its effectiveness.

## REFERENCES

[1] R. Li and Y. G. Li, "Research of COM component based JACOB calls in JAVA," *Microcomputer Information*, 2007.

[2] S. C. Niu and A. Q. Xu, "A COM-based instrument in-terchangeable framework and its realization approach," *IEEE Autotestcon*, 2008.  
 [3] Z. B. Wang and J. H. Han, "Research and Implementation on call COM component of fingerprint identification under Java environment," *Computer Technology and Development*, 2011.  
 [4] H. R. Song, "Research and application of component technology in trademark examination system," *Beijing University of Posts and Telecommunications*, 2012.  
 [5] B. H. Jin, Z. Li, H. Li, S. L. Lv, and Q. Lin, "Design of universal Java WordWriter component based on JACOB," *Journal of Zhengzhou University of Light Industry (Natural Science)*.



**Lizheng Jiang** was born on April 1,1966, in Beijing, China. He received his Ph.D. in computer science and the M.S. in administration science from Peking University. He is now working as a teacher in North China Electric Power University (NCEPU). His research fields include data base, data warehouse, data mining, etc.

He mainly engaged in academic research work and achieved a numbers of achievements. He has published "An efficient cube structure for stream data analysis," "An empirical study on risk data quality management, mining maximal correlated member clusters in high dimensional database," etc. He has lead the development multidimensional electric data stream analysis. Dr. Jiang has been awarded excellent teacher in NCEPU.



**Xin Zheng** was born on January 3, 1991, in Beijing, China. He got his undergraduate degree from North China Electric Power University (NCEPU). He is now studying at Institute of Network and Information Systems in NCEPU to achieve his master's degree. His research fields include mobile internet, network security, cloud computing, etc.

He involved in developing educational administration information distribution system (Eplian-MCIS) and operation and maintenance management system. All of those has obtained software copyright. He has practiced in NEC of China. He is developing energy information distribution system in NCEPU and has published design and implementation of automatic report generation system in academic exchange annual meeting in NCEPU. Mr. Xin has been awarded excellent graduate students in NCEPU and actively participate in school activities.



**Jiantao Zhao** was born on February 2,1963, in Raoyang County, Hebei province, China. He is the associate professor of control and computer engineering and the supervisor of postgraduate in North China Electric Power University (NCEPU). He is the head of the Institute of Network and Information Systems in NCEPU and the chairman of the company EPLIAN. His research fields include internet and mobile internet application software technology, enterprise application software technology, etc.

He hosted and participated in many projects in recent years, include operation and maintenance management system for Qinghai Electric Power Company of the State Grid, digital cloud service platform for science park of NCEPU, energy information distribution system, etc.

Prof Zhao has published more than 30 academic papers and obtained software copyright as much as 16.